AD-A096 541   AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA)   F/G 9/2
ARL HYBRID COMPUTING SYSTEM CONSOLE FUNCTIONS.(U)
MAR 80 H A THELANDER
UNCLASSIFIED   ARL/SYS NOTE-72   NL

1 OF 1
AD A
096541

END
DATE
FILMED
4-81
DTIC

LEVEL II (12)

ARL-SYS-NOTE-72 AR-001-800

AD A096541

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

SYSTEMS NOTE 72

ARL HYBRID COMPUTING SYSTEM CONSOLE FUNCTIONS

by

H. A. THELANDER

Approved for Public Release.

COPY No 20

MARCH 1980

CONTENTS

1. INTRODUCTION

The ARL Hybrid Computing System Mark 3 (HCS3) is the culmination of a facility development initially undertaken in support of guided weapons projects, notably IKARA. Currently HCS3 finds its major employment in the real-time modelling of flight vehicle dynamics for manned flight simulation.

The HCS3 hardware arrangement is shown in Figure 1. The analogue part of the computations is performed by the Modular Analogue Computer (MAC), and the digital part by the ARL DECsystem-10 central timesharing computer. The interface between these two is the PDP-11/20. The hardware is described in more detail below.

The author has described aspects of the predecessor system HCS2 [1, 2]. Significant enhancements have been incorporated into HCS3: these include the provision of Digital Coefficient Units (DCUs), the S-Bus sub-system, better error monitoring and individual integrator mode control, all achieved via the DIGIBUS system [3] on the PDP-11/20. Analogue to digital conversion capacity has been greatly expanded. System software has developed in concept and utility both in regard to hybrid computation [4] in direct comparison with HCS2, and through its provisions for utilization of the capabilities of the Command and Information System (CIS) [5].

In the HCS3 environment, user control and monitoring functions are exercised by interaction with a DECsystem-10 digital program through a timesharing terminal. A class of operations which includes the setting-up and debugging of the analogue parts of hybrid computations, purely analogue computations, and analogue computer hardware maintenance, is best handled with a general purpose digital program for user interaction. The alternative to this is to write and debug a special digital program for each application, an approach which is both inefficient and inconvenient.

This paper describes CONSOL, a program which in HCS3 provides the capabilities necessary for the class of operations mentioned. User commands to CONSOL are input at a DECsystem-10 timesharing terminal, and data, status and error information are returned to the user through the CIS refreshed graphic display.

Chapter 2 provides a broad description of the HCS3 hardware and software sub-systems and its operating environment. The commands and output presentation provided by CONSOL are described in Chapter 3. Implementation is discussed in Chapter 4, and Chapter 5, in conjunction with Appendix A, a pocket reference card, explains the use of CONSOL.

2. HCS3 SYSTEM

2.1 Overview

User programs for performing computations with HCS3 are prepared in two parts: a digital part and an analogue part. The analogue part may be viewed as a fast subroutine called upon by the digital part to perform the high speed dynamic computations at which analogue methods outperform digital. The digital part provides the main interface between the user and his hybrid computation. The digital part has essentially full control of the analogue part.

Programming of the analogue part is done by conventional methods, and the analogue program is implemented with patch wires and modules. The digital part is programmed on the DECsystem-10 using its FORTRAN-10 language [6].

In operation the analogue computer performs its computations continuously, while the digital computations are performed repetitively at a rate controlled by a real time clock. Each cycle of digital computation involves first reading quantities from the analogue part, then performing serial digital computations using those quantities, and finally outputting other quantities to the analogue part. In HCS3 these repetitive digital computations are performed as an interrupt routine, executed in response to regular priority interrupts from the real time clock.

An HCS3 digital part program, whilst syntactically a single entity, actually consists itself of two parts: one part is the normal timesharing part and the other is the repetitively executed real-time part. The HCS3 software provides mechanisms for communication between these two parts. In operation it is not unusual for the timesharing part (of the digital part of a hybrid computation) to be the job scheduled for execution at the time of a real time clock interrupt, and for it thereupon to be suspended during execution of its own real-time part.

The digital program interacts with HCS3, and thereby the analogue program running on the MAC, by means of calls to subprograms in the HCS3 software sub-system H3PAC [4].

2.2 HCS3 Hardware Sub-systems

2.2.1 DECsystem-10

The DECsystem-10 is the ARL central timesharing digital computer. It is configured as a DECsystem-1070 [7], as shown in Figure 2.

The DECsystem-10 I O Bus is connected to the PDP-11 20 by means of the ARL-designed XIX interface (see below).

Operation of the DECsystem-10 is controlled by the TOPS-10 operating system [8], a swapping, virtual-memory timesharing monitor. The HCS3 software sub-systems utilize real time features in TOPS-10.

2.2.2 PDP-11 20

Figure 3 illustrates the main components of the PDP-11 20 sub-system [9, 10]. It includes KA11 central processor, 24K words of 1 μsec core memory, RK11 disk control with two RK05 1·25 m word removable disk cartridge drives, KW11-P programmable real time clock and KE-11A extended arithmetic element. Special devices connect to the PDP-11 20 through DR11-A or -C register interfaces and DR11-B direct memory access interfaces.

The XIX interface uses a DR11-A and provides bi-directional data and interrupt transmission between the DECsystem-10 and the PDP-11 20, in a fashion similar to the DEC DA10 DECsystem-10 to PDP-8 interface [7]. The CIS display controller [5] is interfaced via a DR11-A and provides refreshed directed beam graphics on a 483-mm diagonal cathode ray tube display (Hewlett-Packard 1310A) from display code generated in the DECsystem-10 and stored in the memory of the PDP-11 20.

The PDP-11 20 controls an analogue to digital conversion sub-system comprising a 512-channel input multiplexer (MUX) and 14-bit analogue to digital converter (ADC). Two DR11-B direct memory access interfaces are used with its controller, one for transmitting MUX addresses, the other for receiving ADC data. Conversion full scale ranges of 10 or 100V and automatic overrange detection are program controllable. Over 30,000 conversions per second from random MUX addresses and scales can be made.

The other device utilizing a DR11-B direct memory access interface is the controller for the DIGIBUS, a serial bi-directional data bus [3] which extends through the racks of the MAC. The digital coefficient units (DCUs), digital to analogue converters (DACs), computing element operational states and discrete logic signals in the MAC are controlled by the DIGIBUS. MAC error conditions are monitored by it.

Operation of the PDP-11 20 is controlled by the program CISHCS (see below): no proprietary operating system or executive is used, and the HCS3 hybrid computation user has no access to the PDP-11 20.

2.2.3 Modular Analogue Computer

The MAC, illustrated in Figure 4, is a 100 V scale solid state fixed patch computer. It consists of ten identical computing racks, numbered 0 to 9, and rack 10, used for input and output. Each computing rack contains 40 operational amplifiers, allocated as four fields of ten. Amplifiers 00 to 09 are summers which may, with the insertion of appropriate transfer function (TF) modules in reserved locations, be converted to other functions, for example multipliers,

trigonometric function or variable diode function generators; their operational modes are under common control within each rack. Amplifiers 10 to 19 are front panel switch selectable as summers or integrators with, in the latter case, individual mode controls. Amplifiers 20 to 29 are summers and 30 to 39 inverters. Overload sensing for each amplifier, and oscillation detection for each rack are provided; these error conditions may be read by the DIGIBUS.

Each computing rack can accommodate up to three servo-multipliers or -resolvers, up to thirty coefficient potentiometers, up to thirty digital coefficient units (coefficients in the range 0·00025 to 16·0) set by the DIGIBUS, limiters, relays and comparators. The modular construction of the MAC allows these devices to be installed as and where required.

There are sixteen 16-bit DACs in rack 10, controlled by the DIGIBUS. Each drives one line of the S-Bus sub-system. In each computing rack each of these lines is brought to the normally-open contact of one of sixteen changeover relays, whose normally closed and wiper contacts are brought to a front panel patch. These 160 relays can be individually picked and dropped under control of the DIGIBUS. The S-Bus sub-system is used to verify sub-sections of an analogue program; this is done using the DACs to stimulate their inputs rather than outputs from other parts of the program.

Another DAC drives the IC Cal Bus; it appears on patches throughout the MAC and is available for setting integrator initial conditions.

In each computing rack sixteen discrete logic level outputs controlled by the DIGIBUS are available to the user. These can be individually set and reset by the digital part program.

As well as the 512-channel MUX ADC, to which are hardwired all operational amplifier outputs, the MAC has an access system which, by relay switching under DIGIBUS control, can address 100 points in each rack, including all amplifier outputs and power supplies, patchable inputs in each rack, and integrator inputs, the last through an amplifier whose gain can be set to 0·01, 0·1, 1·0 or 10·0 also under DIGIBUS control.

2.3 HCS3 Software Sub-systems

2.3.1 H3PAC

The main software sub-system in HCS3 is the DECsystem-10 FORTRAN-callable subprogram package, H3PAC. As the user digital program to HCS3 interface it provides the framework within which hybrid programs are designed and implemented. In it are embodied many of the concepts which distinguish HCS3 from other hybrid computation systems [11, 4].

H3PAC is written in MACRO-10 assembler language [12].

Subprograms in H3PAC provide the following capabilities to the hybrid computation digital part program.

(a) Hybrid Program Operation Control

Subprograms in this group allow for hybrid operation initiation and termination, for starting and stopping the iterative execution of the time-critical computations, and for the timesharing part to enter a state of suspension from which it may be woken by its own real time part. This last function is necessary for efficient unambiguous communication between the two parts.

(b) Initialization

In this group are subprograms for setting integrator initial conditions using the IC Cal Bus and for loading function generator tables.

(c) Hybrid Data Manipulation

This group includes subprograms for sending settings to DCUs, for manipulating user discrete logic outputs, and for reading variables from the MAC via the MUX ADC sub-system.

(d) **MAC Control**

H3PAC subprograms in this group allow the digital program to set MAC rack operational states (STAND BY, POT. SET, OP. CAL. or COMPUTE) and individual integrator and rack common operational modes (INITIAL CONDITIONS, RUN or HOLD), and to control integrator timescales.

(e) **Access System**

Subprograms in this group allow the access system address to be set, and the accessed quantity to be read.

(f) **Error Monitoring**

These subprograms return concise or expanded information on MAC errors (overloads, oscillation, and MUX/ADC sub-system hardware overrange detection).

(g) **Data Logging**

HCS3 allows MAC variables to be logged on the RK05 removable cartridge disks, and this group of sub-programs provides control of this function.

(h) **Digital Function Generation**

HCS3 includes facilities for generation of analogue functions of analogue quantities by the PDP-11/20. This group of subprograms controls these facilities.

(i) **S-Bus and IC Cal Bus**

In this group are subprograms for controlling the settings of the S-Bus and IC Cal Bus DACs and the S-Bus Relays.

(j) **Environmental**

This group contains subprograms by means of which the user can examine and set an upper limit to his utilization of the DECsystem-10.

H3PAC contains extensive error detection and reporting capabilities.

2.3.2 LOKXIX

The use of TOPS-10 real-time features by H3PAC requires negation of some system integrity mechanisms. To minimize the resulting risk, and to allow correct sharing of the XIX interface between H3PAC and the CIS and RK05 disk sub-system software packages described below, a MACRO-10 program module LOKXIX is used. LOKXIX enforces a program loading strategy which is designed to protect the software packages using the real-time features, and contains a major part of the code for communication with the PDP-11/20 via the XIX interface. LOKXIX must be loaded with H3PAC or the other real-time software packages; it is separate from them to save run-time memory.

2.3.3 CISPAC and RKPAC

These are two system software modules which are peripheral to HCS3, and are not necessarily used in conjunction with it. They do, however, share use of LOKXIX, see above, and CISHCS, see below, and run in the DECsystem-10.

CISPAC is a package of FORTRAN-callable subprograms, written in MACRO-10 assembler, used for control of the CIS display sub-system. The functions of the CISPAC subprograms [5] are similar to those of the VISTRAN package developed at ARL for use with the DECsystem-10 type 338 interactive display [13]. A number of CISPAC subprograms may be called from the real-time part of an HCS3 digital program.

RKPAC is a package of FORTRAN-callable subprograms, like CISPAC written in MACRO-10, used for transfers of data between a DECsystem-10 program and the RK05 disks of the PDP-11/20. Its main use is in recovery of data logged on the disks by the RDP-11/20 during hybrid computation.

2.3.4 CISHCS

CISHCS is the program which runs in the PDP-11/20 sub-system of HCS3. It is written in the MACRO-11 [14] assembly language, maintained on the DECsystem-10, and cross-assembled by the PALX11 or MACY11 cross-assemblers.

CISHCS controls the PDP-11/20 side of the XIX interface for communication with the DECsystem-10. In HCS3 it performs data formatting and device control functions. It controls the KW11-P programmable real time clock, whose periodic interrupts time the iterative execution of the real-time part of the digital part of the hybrid program in the DECsystem-10. It handles the other HCS3 hardware and performs the digital generation, by table look-up and interpolation, of analogue functions of analogue variables.

For CIS display (CISPAC) and RK05 disk (RKPAC) sub-system functions, CISHCS again performs data transfers and device control.

3. CONSOL COMMANDS AND DISPLAY

3.1 User Interaction

CONSOL is operated from a DECsystem-10 timesharing terminal, at which the user types commands. Its output is displayed on the CIS display.

CONSOL signifies its readiness to accept a command by outputting a "*" (asterisk, the conventional DECsystem-10 program terminal prompt character) at the terminal. Commands by the user are terminated by the "RETURN" key (carriage return, echoed by the DECsystem-10 as 'Carriage Return' 'Line Feed').

Henceforth, in the interests of clarity of exposition, the use of the "RETURN" key to terminate a command to CONSOL is implied and will be omitted.

The only other terminal output from CONSOL is "?" (question mark) which is emitted in response to an erroneous command. Commands may be syntactically erroneous, that is to say unintelligible or ambiguous, or they may be contextually erroneous, for example addressing a device in an unpowered rack or attempting to enter RUN mode when in STAND BY state.

3.2 Commands

CONSOL commands are mnemonic and may be abbreviated according to a logical scheme. In illustration of this point, moving somewhat ahead in the discussion, consider the commands used firstly to set the MAC state to Pot. Set, and secondly to pick up S-Bus Relay 203 (that is the relay in rack 2 on S-Bus line 3). The former is the single letter command "P", and the latter "PICK 203" (where the presence of any non-numeric characters between the I and the 2 is immaterial), which in most abbreviated form becomes just "PI203". The use of the single letter command "P" is not ambiguous in this context, as an S-Bus Relay command requires a numeric device identifier argument.

In the text which follows, minimum abbreviations are given in boldface type, thus **PI**CK**203** could be typed as PICK203, or PIC203, or PI203, but not as P23. Commands are parsed left to right until their alphabetic section is uniquely determined. If a numeric argument or arguments is then required, the remainder of the command line is searched for the first occurrence of a numeric character (or in the case of signed or non-integer numbers the characters + , - , .

. as well), and a number assembled from it and subsequent numeric characters until a non-numeric character (or end of input line) is reached.

In CONSOL commands, space and tabulation characters are valid as delimiter characters in numeric arguments so that "2 3" is equivalent to "2" followed by "3" not "23". Alphabetic characters may be entered in upper or lower case.

An exhaustive listing of CONSOL commands follows.

3.2.1 MAC State Commands

State commands are single letter commands which control the operational state of the entire MAC (those racks, in addition to rack 10, which are powered up). The four state commands are

S

P

O

and **C**

which respectively set the MAC operational state to Standby, Pot.Set, Op.Cal., and Compute. Changing the MAC state from Op.Cal. or Compute back to Pot.Set or Standby resets all operational mode controls to Initial Conditions.

3.2.2 MAC Mode Commands

CONSOL provides operational mode control for the whole MAC by means of "MAC-wide" mode commands. These set all integrator and rack-common modes (in the powered-up racks). As well, there are commands for setting modes of individual integrators and rack commons.

Modes may be set, and indeed they have meaning, only when the MAC is in Op.Cal. or Compute state.

Single letter commands are used for MAC-wide mode control:

I

R

and **H**

which respectively set all MAC modes (in powered-up racks) to Initial Conditions, Run or Hold.

Control of individual integrator or rack common modes is achieved by affixing an argument list to one of these basic commands. The argument list consists of one or more (decimal) integer numbers separated, and if desired separated from the basic command, by any character or characters not from the set of ten decimal digits. These integer numbers specify the rack and unit numbers of the devices whose modes are to be set. Rack common modes are given unit number 20 for this purpose only.

Thus, the CONSOL command

RUN 210, 211, 212, 315, 320

will, provided the MAC is in Op.Cal. or Compute state, put integrators 10, 11 and 12 in Rack 2, integrator 15 in Rack 3, and the Rack 3 common mode control all into Run.

3.2.3 MAC Access System Commands

CONSOL displays the value of the variable selected by the MAC access system, averaged to minimise mains frequency noise and updated approximately once per second. Access address commands, for selecting the various access points in the MAC, consist of one of the letters

A

P

D

or **S**

followed by an integer number. Respectively, these select Amplifiers, Power supply rails, Differential Check points and Special (rear-patchable) points; the integer number specifying the rack and unit. MAC computing racks contain 40 amplifier positions (numbered 00 to 39), 20 of which (00 to 19) have differential check access, ten power supply access points (00 to 09), and thirty (00 to 29) special access points.

Thus

A011

selects amplifier 11 in rack 0 for access,

A101

selects amplifier 1 in rack 1, and

A1001

accesses amplifier 1 in rack 10.

Differential check access requires that the MAC be in Compute or Op.Cal. state. The differential is read using an amplifier whose gain can be varied under program control (see 2.2.3 above) in decades from 0·01 to 10·0. CONSOL uses the highest gain setting that gives a within-range reading at the amplifier output.

As mentioned below, the display of the access output is always in terms of machine units (M.U.).

The allocation of power supply units and the amplifiers in Rack 10 are given in Appendix B.

3.2.4 Digital Coefficient Units

Unlike the coefficient potentiometers which they are tending to supplant, DCUs do not retain their settings, and so in many problem set-up and hardware maintenance activities there is a need to be able to transmit values to one or more DCUs. CONSOL provides this capability with a command of the form

DCU *list*

where *list* is a list containing zero or more DCU address data specifications, separated by any characters not legal within a DCU address data specification. In turn, a DCU address data specification takes the form of an integer number specifying the rack and position in that rack of a particular DCU, separated by one or more characters not including the ten decimal digits from a decimal number whose value lies in the range 0 to 16·0, specifying the coefficient to be sent to that DCU.

If *list* is empty, all DCU coefficients set up by previous commands are reset to zero.

Thus the command

DCU 005 : 12·1, 107 : 0·04, 113 : 6

would set coefficients of: 12·1 on DCU 5 in Rack 0, 0·04 on DCU 7 in Rack 1, and 6·0 on DCU 13 in Rack 1. The command

DCU

alone would clear all coefficient settings.

3.2.5 S-DACs and S-Bus Relays

The S-Bus DACs Relays provide HCS3 with a useful problem subset checkout facility. Important input signals can be patched through the normally closed contacts to the wipers of the S-Relays. Test programs, by switching the relays, can select the S-DAC outputs (through the normally open S-Relay contacts) instead of those signals to stimulate computational elements.

CONSOL has commands allowing any of the S-Relays to be "picked" from their quiescent state, or "dropped", and values to be output to the S-DACs. The relays are controlled by the commands

PICK *list*

and **DROP** *list*

in which *list* contains S-Relay address specifications in the form of decimal integers separated by any non-numeric characters. S-Relay addresses consist of a single digit rack number followed by a two digit Bus-line number. The command

DROP

with no argument list results in all S-Relays currently picked being dropped. For example, the command

PICK 203, 300, 002, 005

will result in S-Relays 3 in Rack 2, 0 in Rack 3, and 2 and 5 in Rack 0, all being picked.

Values are sent out to the S-DACs using commands of the form

SDAC *list*

where *list* contains zero or more S-DAC device data specifications, separated by any characters not legal within an S-DAC device data specification. An S-DAC device data specification in turn consists of a decimal integer DAC number ranging from 0 to 15 (which may be preceded by any number of zeroes), separated (by any non-numeric characters) from a decimal number in the range −1·0 to +1·0 giving the value to be sent to that particular DAC.

If *list* is empty, all S-DACs are set back to zero output.

For example, the command

SDAC 0 : 0·1, 13 : −0·5

will send the value 0·1 to S-DAC 0, and −0·5 to S-DAC 13.

3.2.6 Initial Conditions Bus DAC

The MAC IC Cal Bus is used in calibration, as explained below, and also for setting up integrator initial conditions values, when it is often referred to as simply the IC Bus. CONSOL can set values on the bus using the command

ICBUSDAC *value*

where *value* is a number in the range −1·0 to +1·0 specifying the value to be set on the bus. If it is omitted, zero is assumed.

To set an initial condition value to an integrator (whose IC input is patched to the IC Cal Bus), the MAC must be in Op.Cal. or Compute state and the particular integrator in Initial Conditions mode. The desired value (sign-changed to account for the integrator inversion) should be sent to the IC Bus DAC. After one second the integrator will settle to this value and can be put into Hold mode while others are set.

3.2.7 User Logic Bits

CONSOL allows setting and clearing any of the sixteen discrete user logic signals in any of the MAC racks with the commands

SET *list*

and **CLEAR** *list*

in which *list* contains bit addresses in the form of decimal integers separated by any non-numeric characters. A bit address consists of a rack number (0–10) followed by a two digit bit number (00–15).

If the CLEAR command is given with no argument list, all bits are cleared.

3.2.8 Calibration

An important function of CONSOL for hybrid and analogue computing use is its provision

for calibrating manual coefficient potentiometers. An analogue null indication on the CIS display is provided by CONSOL for calibrating.

To set a coefficient the command

CAL *amp : value*

is used. Here *amp* is the address of the amplifier to which the coefficient potentiometer is patched, expressed in the integer form used in access address commands (see 3.2.3 above); the delimiter, shown here as colon may be any non-numeric character or characters, and *value* is the desired coefficient setting, a positive number which may range up to 1000. The CONSOL calibration function can not be used to set coefficients greater than 1000, but these are rare.

Calibration requires that the MAC be in Pot.Set. or Op.Cal. state, if it is not CONSOL treats the calibration command as an error.

In calibration, the lower, moving null scale pointer should be brought into coincidence with the upper fixed pointer. The access value reading on the display indicates the coefficient setting achieved. Details of the calibration process and pointer mechanization are discussed in Chapter 4.

3.2.9 Reference Card

A CONSOL Reference Card, summarizing the above information, forms Appendix A to this Note. Spare cards are available.

3.3 Display

Figure 5 is a photograph of the CIS display obtained when running CONSOL, and Figure 6 is an annotated sketch of the display, to which reference will be made in the following subsections.

3.3.1 State, Mode and Error Displays

At the top left of the CONSOL display is the indicator of the MAC operational state, item (a) in Figure 6. In the top centre is the indicator of the MAC-wide operational mode, item (b). This shows the mode selected by the last MAC-wide (see 3.2.2 above) mode command. At the top right is an error summary area, item (c), in which are displayed indications of error conditions existing somewhere in the MAC. Separate lines are reserved for indications of amplifier overload, amplifier oscillation, servo multiplier resolver protection trip-out, and MAC calibration status. The last may indicate simply that some device in the MAC is undergoing calibration, which is not of itself necessarily an error; that more than one device is in calibration, this is Multiple Cal. and is always an error; or that the MAC is in Compute state with a device in calibration, the Cal. Interlock error.

3.3.2 Access Display

In the upper centre of the CONSOL display, item (d) in Figure 6, is a display of access address, value and noise. The address is displayed in the same form as is used by CONSOL access address commands (see 3.2.3 above), a letter indicating device type followed by an integer device rack and unit number.

Below the address is the accessed value. This is given in machine units (M.U.), except when calibrating (see 3.2.8 above), when the value displayed is that of the coefficient setting achieved (the accessed value scaled according to the IC Cal Bus setting at the time). CONSOL selects the optimum scale for reading and display of the accessed value. Values greater than 0·1 M.U. in magnitude are read using a direct ADC channel, with a full scale range of ± 1·05 M.U., and are displayed to five significant figures, four decimal places. Values whose magnitudes lie between 0·001 and 0·1 M.U. are read using a direct ADC channel with a full scale range of ± 0·105 M.U., and are displayed to five decimal places. Values smaller in magnitude than 0·001 M.U. are scaled by an amplifier of gain 100 and read to a full scale of ± 0·105 M.U.,

giving an effective full scale of 0·00105 M.U., and are displayed to six decimal places (the least significant digit being hundreds of microvolts).

The third line of the access display gives the average of the absolute value of the a.c. component of the selected signal in machine units. It is displayed at a lower intensity than the address and value.

3.3.3 Calibration Null Scale

During calibration, (see 3.2.8 above), CONSOL displays an analogue null indication, item (e) in Figure 6. The fixed scale shows low and high sides, and zero is indicated by a triangle. A moving triangle below this indicates the current setting.

3.3.4 Tabulation

The lower part of the CONSOL display is a tabulation, items (f), (g), and (h) in Figure 6. The top line, item (f), contains the numbers of the powered-up racks and blanks where racks are switched off. In the columns below the rack numbers are rack error indicators, item (g), indicating the presence of amplifier overload, oscillation, servo trip-out and device calibration in each rack. Below these, the columns, item (h), list the operational modes of the integrators and common mode controls in the racks.

4. IMPLEMENTATION

CONSOL consists of a FORTRAN main program with seven FORTRAN subprograms, operating under HCS3 (see Chapter 2). System software packages used are the load checking and XIX communication module LOKXIX, the hybrid package H3PAC, and the CIS display package CISPAC.

The main program contains a timesharing part and a real-time part, and these are described below.

4.1 CONSOL Main Program Timesharing Part

The timesharing part of CONSOL initializes the program, determining which MAC racks are powered-up, clearing data arrays, and setting up much of the display. On completion of these processes, the timesharing part starts up the regular iterations of the real-time part.

The timesharing part performs the terminal interaction with the user, returning repeatedly to this same point. It outputs the command prompt character to the terminal and then enters a wait state from which it emerges when the user terminates a command to CONSOL by striking the carriage return key.

The command line is read one character per element right justified into an input buffer array. Alphabetic case conversion is performed, and characters are counted: if the line is empty or contains only spaces it is ignored and the program loops back to send the prompt character once again. The command is decoded to a first stage by means of a dispatch calculated from the value of the first non-blank character in the command. Further decoding, using the second or subsequent characters is done if needed. Numerical arguments are assembled in integer or floating form using FORTRAN subprograms, as required by the context.

Once the command has been interpreted and its validity checked, the timesharing part determines a function code which it stores for the real-time part. Should the command contain an error, it is ignored, and the command error character is output on the terminal before the command prompt. Ten function codes are produced: for setting access address, MAC state and device mode, for setting and clearing user logic bits, for performing coefficient calibration, for sending values to DCUs, S-Bus DACs and the IC/Cal Bus DAC, and for controlling the S-Bus relays.

4.2 CONSOL Main Program Real-Time Part

The real-time part of CONSOL performs the functions specified by the function codes set up as a result of command decoding by the timesharing part, and maintains the CIS display.

4.2.1 Real-Time Part Cycle

A real-time part iteration period of 82 milliseconds is used by CONSOL. Consider an arbitrary iteration commencing at time T_0. Subsequent iterations occur at $T_0 + 82$ msec, $T_0 + 164$ msec, . . . Clearly any ten consecutive iterations occur at uniformly spaced phases of $2\pi/10$ radians of an arbitrary, coherent 50 Hz signal. The mean of ten readings of the access value made in consecutive real-time part iterations will therefore have any 50 Hz fundamental noise component suppressed. The period of 82 milliseconds was selected as a compromise between the CONSOL user's requirement that the display be updated frequently, implying a short period, and DECsystem-10 processor utilization which varies inversely with the period.

Twelve real-time part iterations make up a complete CONSOL program cycle, taking 984 milliseconds. This period is manifested in the slight blink exhibited by the access value displayed by CONSOL. This blink provides a useful indication that the program is running when the display content is otherwise stable.

During the first real-time part iteration of the cycle of twelve, the program tests whether there has been a function code stored, in response to a user command, by the timesparing part. If there has not, the iteration terminates. If a function code has been stored, the program uses its value to dispatch to the appropriate function handler code where commands, data, or both are constructed and transmitted to the MAC. The stored function code is then cleared and the iteration terminated.

In the second real-time part iteration the program takes no action, allowing any commands or data to propagate into the MAC. This idle iteration is necessitated by the HCS3 timing characteristics, described more fully in the H3PAC Manual [4], which provide that commands and data directed to the MAC during a real-time part iteration are held in the PDP-11/20 and forwarded after completion of data input at the start of the next real-time part iteration. This strategy allows maximum time for transients resulting from the commands or data sent to the MAC to decay before MAC data acquisition begins.

The remaining ten real-time part iterations of the cycle are used for reading the data from the MAC. These values are summed and from the sums average values are computed for display.

4.2.2 Function Handlers

CONSOL maintains a set of data reflecting the current status of the MAC. This permits the terminal interaction sections of the CONSOL timesharing part to discard redundant commands, saving processing, and to detect contextual errors, for example requesting Run mode in Standby state. When, in the first real-time part iteration of a cycle of twelve, a function code set up by the timesharing part is encountered, a function handler is invoked. These handlers use calls to H3PAC subprograms to send commands and data to the MAC. On successful completion of these transfers, the MAC status data are brought up to date.

4.2.3 Display Content Updating

The computation load deriving from updating the CONSOL display is spread by performing part of it in each of the twelve real-time part iterations of a cycle. When a large part of the display content is varying, over a hundred milliseconds of DECsystem-10 processor time may be consumed in updating the display. If this were not spread over the cycle a real-time part iteration period of several hundred milliseconds would be required to avoid the operationally fatal real-time overrun error which occurs when a real-time part iteration has not terminated at the time that the next one should commence.

On the display the propagation of a MAC-wide mode command through the tabulation of integrator modes illustrates this load spreading. CONSOL maintains internal data reflecting current display content, allowing redundant display code to be suppressed.

For speed, the conversion of the access value from numeric form (in which it is obtained from H3PAC) to the stream of ASCII characters required by the subprograms of CISPAC is done directly with FORTRAN arithmetic rather than by the FORTRAN ENCODE statement. Again for speed, the multiple display of the access address and values needed for adequate brightness of the large characters is achieved by generating them as single sub-pictures [5] and calling them repeatedly.

4.2.4 Coefficient Calibration

The CONSOL coefficient calibration function, see 3.2.8 and 3.3.3 above, differs slightly from the other CONSOL functions in that it is a dynamic process which remains active until terminated by a subsequent command input.

In the first real-time part iteration of the first cycle following a calibration command, the access system address of the amplifier under calibration is set up and an initial calibration signal value is computed and sent to the IC Cal Bus DAC. For coefficients smaller than 1·0 this value is 1·0 M.U., for coefficients between 1·0 and 10·0 it is 0·1 M.U., for coefficients between 10·0 and 100·0 it is 0·01 M.U., and for coefficients between 100·0 and 1000·0 it is 0·001 M.U.

In the case of coefficients greater than 1·0, that is for IC Cal Bus DAC settings smaller than 1·0 M.U., the program waits until the achieved coefficient setting comes within three percent of the coefficient value specified in the command. In the first real-time part iteration of the next cycle the value sent to the IC Cal Bus DAC is altered to the reciprocal of the specified coefficient, so that the output of the amplifier under calibration is 1·0 M.U. at the desired value. This allows the full accuracy of the ADC to be used.

During calibration the value displayed is the achieved coefficient. This in turn is the access value multiplied by the reciprocal of the value currently set on the IC Cal Bus DAC.

The analogue null indicator displayed during calibration has a highly non-linear mechanization of the form

$$x = k \sin\left(\frac{\pi}{2} \sin\left(\frac{\pi (C_a - C_s)}{2 (C_a + C_s)}\right)\right),$$

where its position x is 0 at null, k is a screen-size scale factor, C_a is the achieved and C_s the specified coefficient. The term $(C_a - C_s)/(C_a + C_s)$ has the desirable characteristic of taking the values 0 when $C_a = C_s$, -1 when $C_a = 0$, and $+1$ when $C_a = \infty$, for $C_s \neq 0$. The two applications of the sin function each increase the slope of the function at null, and hence the indicator's sensitivity, by a factor of $\pi/2$.

4.3 CONSOL Subprograms

Two of the subprograms used by CONSOL have already been mentioned: these are used by the timesharing part to scan the input command line to extract the next integer or floating format number.

A subprogram is used by the timesharing part in the initial setting up of the display pictures. Another is used to convert device addresses given in mode commands to the form used by H3PAC. Subprograms are used to perform element-by-element comparison of arrays and to test for numeric characters.

Only one FORTRAN subprogram is used by the real-time part. Its function is to expand the MAC error data from the machine-dependent form in which it is obtained from H3PAC into a form from which the CONSOL display MAC error tabulation can be constructed.

5. OPERATING ENVIRONMENT

The CONSOL program is kept in saved form on the Flight Systems Group library disk area. FSG users can run the program by typing the command

RUN CONSOL

to the DECsystem-10 Monitor's prompt character ".", after logging in. Other users should seek advice from FSG.

Exit from CONSOL is by means of the command

EXIT

which is preferred to use of the control-C mechanism because it clears the CIS display.

CONSOL is not known to contain any errors, any that are discovered and that are reasonably repeatable should be reported to the author.

Rarely, CONSOL will fail with the typeout of an error message from H3PAC specjfying

REAL TIME ERROR -- OVER RUN.

This occurs because of the compromise value selected for the program's real time iteration period, which must be as small as possible while allowing time for all CONSOL's activities. Heavy DECsystem-10 Magnetic Tape system usage is the most frequent cause. The program may be run again with safety after this type of failure.

It will be observed that there is a distinct pause between the issuance of the command to run CONSOL and the appearance of its display on the CIS. This is an unavoidable consequence of real time operation initialization.

REFERENCES

1. Thelander, H. A., "A Time-sharing Digital Computer in Real-Time Hybrid Simulation", Proceedings of the Digital Equipment Computer Users' Society, Vol. 1, No. 5, Melbourne, August 1975.
2. Thelander, H. A., "Hybrid Computing System II Digital Programming Manual", Systems Factors Group Memorandum SFG 23, ARL, September 1973.
3. Kinsella, T., "Specification and Device Design Information for the M.A.C. Serial Digital Bus System", Systems Factors Group Memorandum SFG 41, ARL, June 1976, Rev. 2, November 1978.
4. Thelander, H. A., "Preliminary Report on the System Software for HCS3", unpublished, ARL, May 1977.
5. Warne, D. K., "Command and Information System 1, Programming and Operation Manual", Systems Factors Group Memorandum SFG 31, ARL, October 1974.
6. ———, "DECsystem-10 FORTRAN-10 Language Manual", Digital Equipment Corporation Publication DEC-10-LFORA-D-D, Second Edition, November 1975.
7. ———, "DECsystem-10 Hardware Reference Manual", Digital Equipment Corporation Publication DEC-10-XSRMA-A-DN1, Fourth Edition, March 1976.
8. ———, "Tops-10 Software Notebooks", Digital Equipment Corporation, December 1978.
9. ———, "PDP-11/20/15/R20 Processor Handbook", Digital Equipment Corporation, 1972.
10. ———, "PDP-11 Peripherals Handbook", Digital Equipment Corporation, 1972, 1976.
11. Thelander, H. A., "HCS3 System Software", In preparation, ARL, Report 1980.
12. ———, "MACRO Assembler Reference Manual", Digital Equipment Corporation Publication AA-C780C-TB, April 1978.
13. Neil, B. D. L., "VISTRAN", ARL Computer Group Bulletin No. 27, May 1976.
14. ———, "Macro-11 Reference Manual", Digital Equipment Corporation Publication DEC-11-OIMRA-A-D, December 1975.

APPENDIX A

CONSOL Reference Card

(Inside Back Cover)

APPENDIX B

Power Supply and Rack 10 Amplifier Allocation

Allocation of the Power Supply Access points in each MAC Rack is shown in Table B-1.

TABLE B-1

Power Supply Access Allocation

Access Unit	*Supply Rail*
00	Analogue Ground
01	Reference Supply
02	Reference Supply
03	125 V Amplifier Power Supply
04	125 V Amplifier Power Supply
05	15 V General Purpose Supply
06	15 V General Purpose Supply
07	5 V Logic Power Supply
08	IC/CAL Bus
09	Spare (unused)

Rack 10 contains twenty amplifiers, which may be selected with the MAC Access System. Their use is shown in Table B-2.

TABLE B-2

Rack 10 Amplifier Allocation

Amplifier	Function
0	S-Bus DAC 0 Amplifier
1	S-Bus DAC 1 Amplifier
.	.
.	.
15	S-Bus DAC 15 Amplifier
16	IC/Cal Bus DAC Amplifier
17	Differential Check Access Amplifier
18	ve Reference Supply Amplifier
19	ve Reference Supply Amplifier

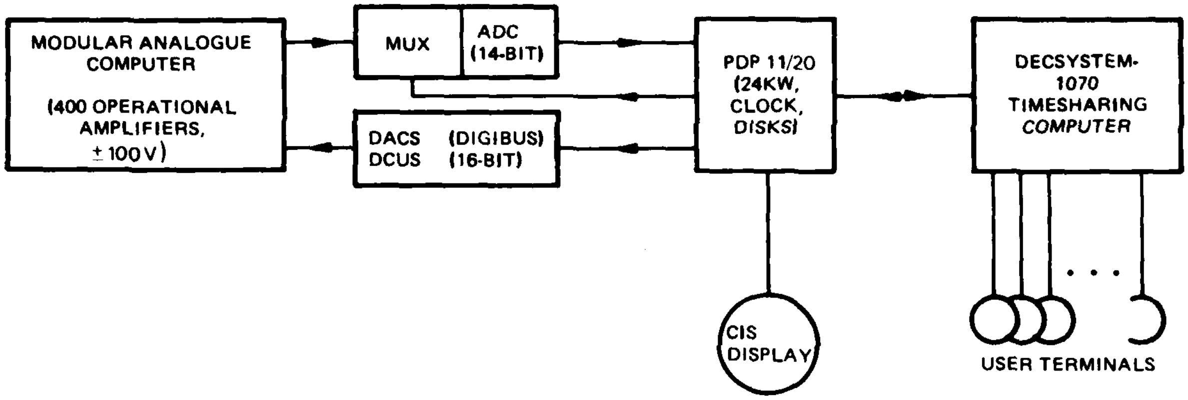

FIG. 1: HCS3 HARDWARE OUTLINE

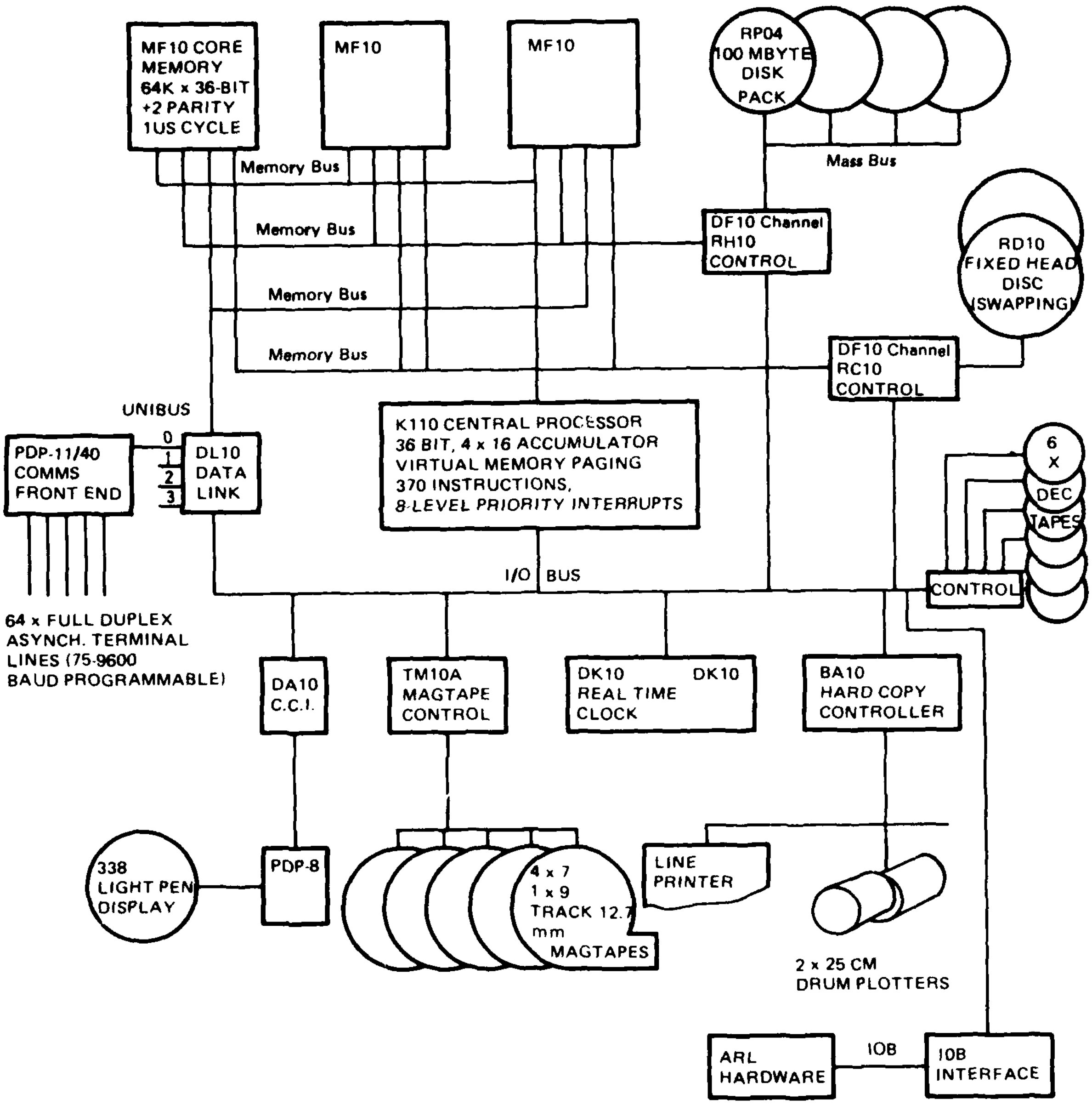

FIG. 2: DECSYSTEM – 1070

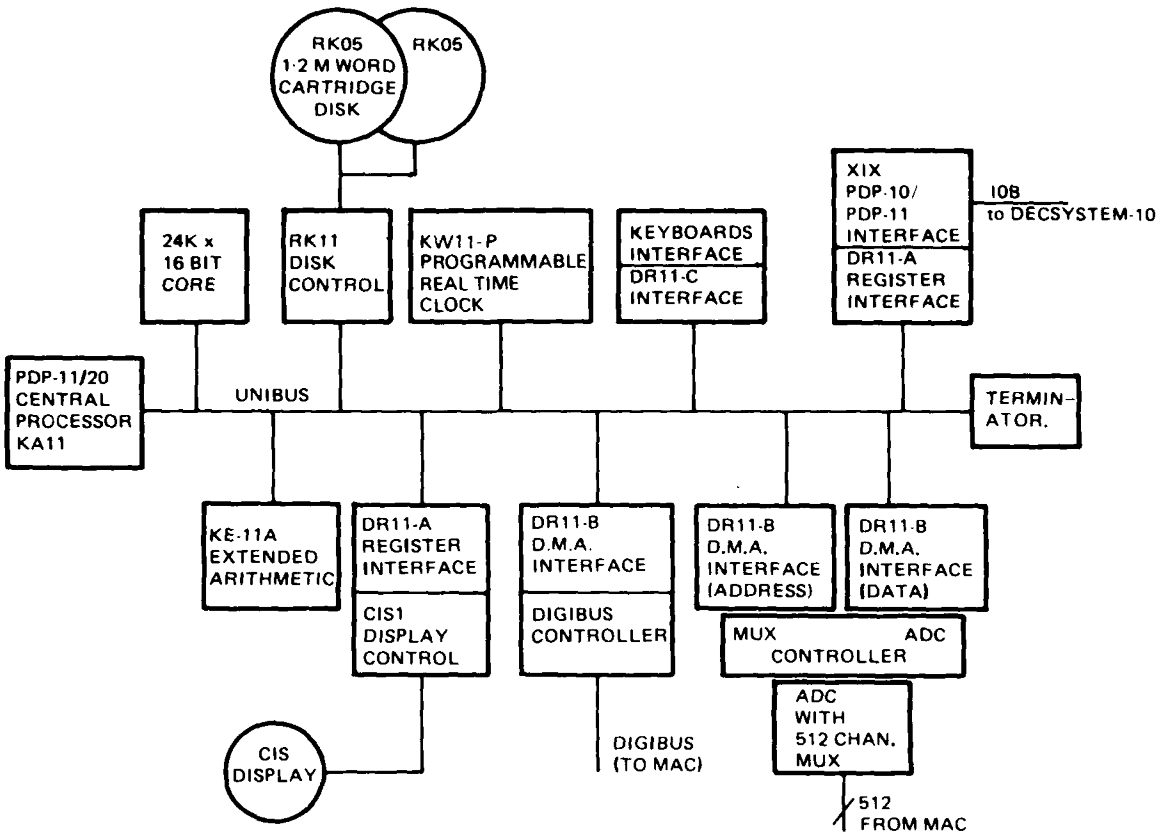

FIG. 3: PDP-11/20 HARDWARE

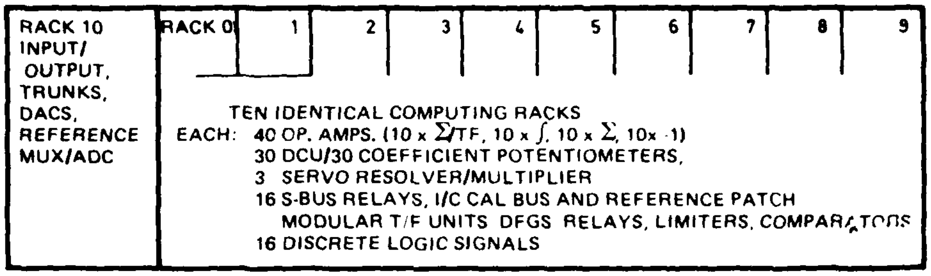

FIG. 4: MODULAR ANALOGUE COMPUTER

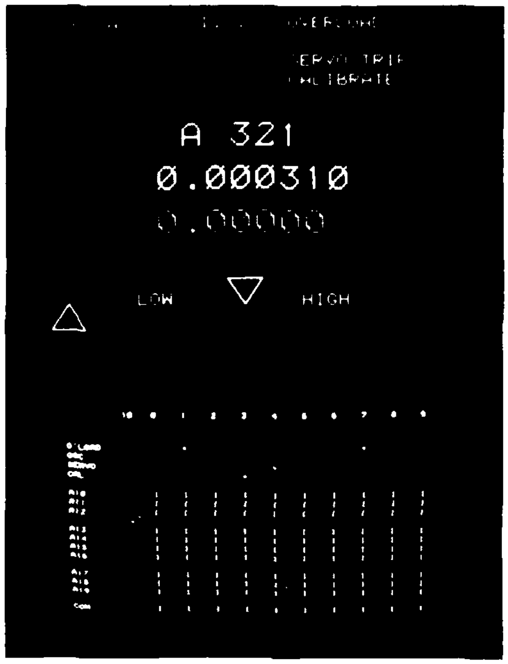

FIG. 5: PHOTOGRAPH OF CONSOL DISPLAY

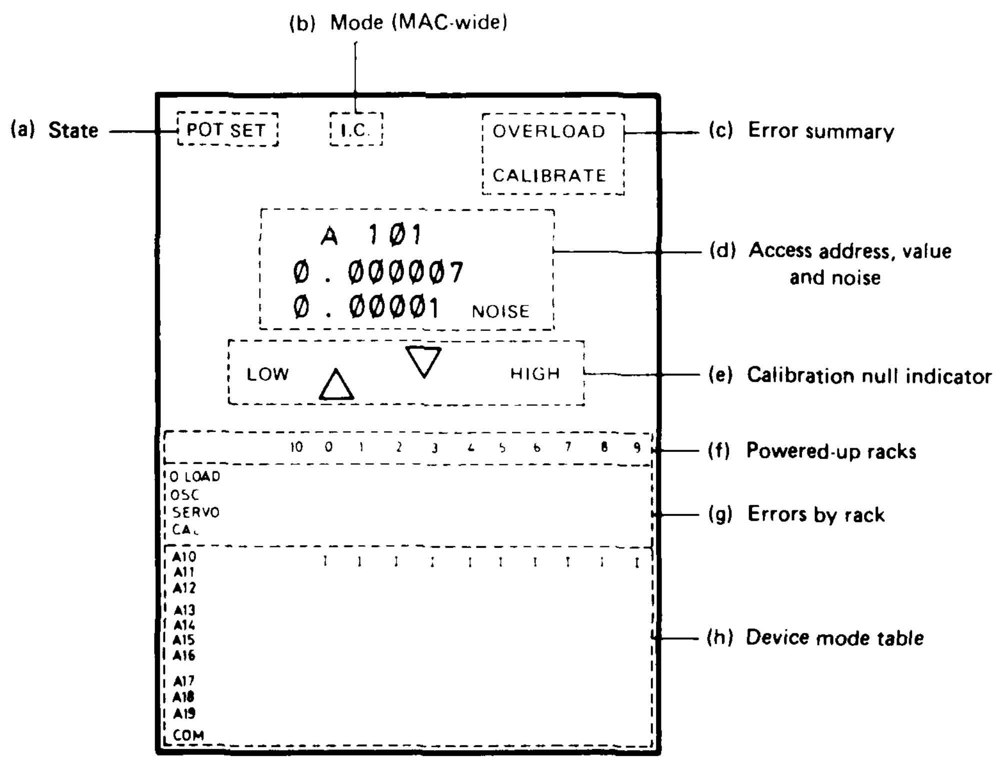

FIG. 6: ANNOTATED SKETCH OF CONSOL DISPLAY

DISTRIBUTION

	Copy No.
AUSTRALIA	
Department of Defence	
Central Office	
Chief Defence Scientist	1
Deputy Chief Defence Scientist	2
Superintendent, Science and Technology Programs	3
Australian Defence Scientific and Technical Representative (UK)	
Counsellor, Defence Science	
Joint Intelligence Organisation	4
Defence Library	5
Document Exchange, D.I.S.B.	6-22
Aeronautical Research Laboratories	
Chief Superintendent	23
Library	24
Superintendent Systems Division	25
Divisional File Systems Division	26
PO FSG	27-32
Author: H. A. Thelander	33
J. Henshall	34
T. Kinsella	35
G. Saleeba	36
J. Sandor	37
K. Rogers	38
W. Wilson	39
Simulation Lab.	40
Materials Research Laboratories	
Library	41
Defence Research Centre, Salisbury	
Library	42
Navy Office	
Naval Scientific Adviser	43
Army Office	
Army Scientific Adviser	44
Air Force Office	
Air Force Scientific Adviser	45
Department of Productivity	
Government Aircraft Factories	
D. G. Bryant	46
Department of Transport	
Library	47
Statutory, State Authorities and Industry	
Australian Atomic Energy Commission, Director	48
CSIRO, Division of Computing Research	49

FRANCE

ONERA, Library		50
Service de Documentation, Technique de l'Aeronautique		51
SWEDEN		
Research Institute of the Swedish National Defence		52
UNITED KINGDOM		
Royal Aircraft Establishment, Farnborough, Library		53
National Physical Laboratories, Library		54
Universities and Colleges		
Manchester	Library, U.M.I.S.T.	55
Cranfield Inst. of Tech.	Library	56
Imperial College	The Head	57
UNITED STATES OF AMERICA		
NASA Scientific and Technical Information Facility		58
Battelle Memorial Institute, Library		59
Universities and Colleges		
Johns Hopkins	Applied Physics Lab.	60
Pennsylvania State	Dept. of Electrical Engineering	61
Spares		62-71

CONSOL

SIDE 1

REFERENCE CARD
HCS III Console Program

DEPARTMENT OF DEFENCE
AERONAUTICAL RESEARCH LABORATORIES
Appendix A to Systems Note 72
H. A. Thelander
March 1980

Notes:

(i) All command lines are terminated with "Carriage Return".
(ii) Erroneous commands are indicated by "?" and ignored.
(iii) Minimum abbreviations are given in bold type.
(iv) Alphabetics may be upper or lower case.
(v) Access value display is always in Machine Units.

To Run the Program:
Type **RUN CONSOL**

To Exit the Program:
Type **EXIT**

MAC STATE CONTROL
Single letter commands
S (Stand by) **P** (Pot. Set) **O** (Op. Cal.) **C** (Compute).

MAC MODE CONTROL (Requires Op. Cal. or Compute State)
MAC — Wide single letter commands
I (Initial Conditions) **R** (Run) **H** (Hold).

Individual Modes
Use the same initial letter followed by *list* of device numbers of form *stu* (where *s* is rack and *tu* $10 \leqslant tu \leqslant 20$, Common Mode is unit 20) separated by non-numerics.

CONSOL REFERENCE CARD

SIDE 2

ACCESS SYSTEM ADDRESS

Initial letter of device type then device (rack and unit number.

A*rstu* (Amplifier) **P***rstu* (Power Supply Rail)

S*rstu* (Special Access) **D***rstu* (Differential Check: Requires Op. Cal. or Compute).

rs is rack number, *r* if zero may be omitted, *tu* is unit number.

DIGITAL COEFFICIENT UNITS

DCU *list*

list contains DCU (rack and unit) numbers each followed by its coefficient.

DCU

alone clears all DCU's.

S-BUS RELAYS AND DACS

PICK *list* **DR**OP *list* **DR**OP (all)

list contains Relay (rack and unit) numbers.

SDAC *list*

list contains S-DAC numbers each followed by its setting. Empty *list* clears all.

IC/CAL BUS DAC

ICBDAC *value*

DISCRETE USER LOGIC BITS

SET *list* **CL**EAR *list* **CL**EAR (all)

list contains bit addresses.

CALIBRATION (Requires Pot. Set or Op. Cal. State)

CAL *amp : value*

Cards available from A.R.L., GPO Box 4331 Melbourne

DAT
FILM